Solving hybrid optimisation problems over real with ASP

Kerian Thuillier

Univ Rennes, Inria, CNRS, IRISA
Rennes, France

23th June 2022

Optimisation problems

examples: linear programming, integer linear programming

$$\begin{aligned}&\text{maximize} \qquad f(x_1, \cdots, x_k)\\&\text{under constraints:}\\&g_i(x_1, \cdots, x_k) \leq 0\\&h_j(x_1, \cdots, x_k) = 0\end{aligned}$$

with $f, g_i, h_j : \mathbb{R}^k \rightarrow \mathbb{R}$

Valid solution: variable assignments satisfying all the inequalities and equalities constraints

Optimal solution: valid solution maximizing the objective function

Optimisation problems

examples: linear programming, integer linear programming

$$\text{maximize} \qquad f(x_1, \cdots, x_k)$$
$$\text{under constraints:}$$
$$g_i(x_1, \cdots, x_k) \leq 0$$
$$h_j(\underbrace{x_1, \cdots, x_k}_{\text{real variables}}) = 0$$

with $f, g_i, h_j : \mathbb{R}^k \to \mathbb{R}$

Valid solution: variable assignments satisfying all the inequalities and equalities constraints

Optimal solution: valid solution maximizing the objective function

Optimisation problems

examples: linear programming, integer linear programming

$$\text{maximize} \quad f(x_1, \cdots, x_k)$$ } Objective functions

$$\text{under constraints:}$$

$$g_i(x_1, \cdots, x_k) \leq 0$$

$$h_j(\underbrace{x_1, \cdots, x_k}_{\text{real variables}}) = 0$$

with $f, g_i, h_j : \mathbb{R}^k \to \mathbb{R}$

Valid solution: variable assignments satisfying all the inequalities and equalities constraints

Optimal solution: valid solution maximizing the objective function

Optimisation problems

examples: linear programming, integer linear programming

$$\text{maximize} \quad f(x_1, \cdots, x_k)$$ Objective functions

under constraints:

$$g_i(x_1, \cdots, x_k) \leq 0$$ Set of inequalities constraints

$$h_j(x_1, \cdots, x_k) = 0$$ Set of equalities constraints

($x_1, \cdots, x_k$: real variables)

with $f, g_i, h_j : \mathbb{R}^k \rightarrow \mathbb{R}$

Valid solution: variable assignments satisfying all the inequalities and equalities constraints

Optimal solution: valid solution maximizing the objective function

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\textbf{ASP constraint}}$$

$H, A_i : p(t_1, \cdots, t_k)$ Atoms composed of a function symbol and a set of terms

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\text{ASP constraint}}$$

$H, A_i : \overbrace{p(t_1, \cdots, t_k)}^{\text{Boolean value } \{0, 1\}}$ Atoms composed of a function symbol and a set of terms

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\textbf{ASP constraint}}$$

Boolean value {0, 1}

$H, A_i : p(t_1, \cdots, t_k)$ Atoms composed of a function symbol and a set of terms

or

$H, A_i : \begin{array}{l} g(x_1, \cdots, x_k) = 0 \\ g(x_1, \cdots, x_k) \leq 0 \end{array}$ Equalities or inequalities constraints

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\textbf{ASP constraint}}$$

Boolean value {0, 1}

$H, A_i : p(t_1, \cdots, t_k)$ Atoms composed of a function symbol and a set of terms

or

$H, A_i : \begin{matrix} g(x_1, \cdots, x_k) = 0 \\ g(x_1, \cdots, x_k) \leq 0 \end{matrix}$ Equalities or inequalities constraints

$H :$ An objective function $\text{maximize} \quad f(x_1, \cdots, x_k)$

Integrity constraints $\forall x_1, \cdots x_k, \; h(x_1, \cdots, x_k) \leq 0$

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\textbf{ASP constraint}}$$

Boolean value {0, 1}

$H, A_i : p(t_1, \cdots, t_k)$ Atoms composed of a function symbol and a set of terms

or

$H, A_i :$ $g(x_1, \cdots, x_k) = 0$, $g(x_1, \cdots, x_k) \leq 0$ Equalities or inequalities constraints

$H :$ An objective function $\text{maximize} \quad f(x_1, \cdots, x_k)$

Integrity constraints $\forall x_1, \cdots x_k, \; h(x_1, \cdots, x_k) \leq 0$

Optimisation theory atoms

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP[1]

Hybrid problems merge constraints of two differents theories

example: combinatorial + linear constraints

$$\underbrace{H \leftarrow A_1, \cdots, A_n, \neg A_{n+1}, \cdots, \neg A_m.}_{\textbf{ASP constraint}}$$

Boolean value {0, 1}

H, A_i : $p(t_1, \cdots, t_k)$ Atoms composed of a function symbol and a set of terms

or

H, A_i : $g(x_1, \cdots, x_k) = 0$, $g(x_1, \cdots, x_k) \leq 0$ Equalities or inequalities constraints

H : An objective function $\text{maximize} \quad f(x_1, \cdots, x_k)$

Integrity constraints $\forall x_1, \cdots x_k,\ h(x_1, \cdots, x_k) \leq 0$

Boolean value {0, 1} — Optimisation theory atoms

[1] C. Baral, **Cambridge University Press**, 2003

Hybrid problems with ASP

Example

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},$$
$$y \leq 0.6$$

Hybrid problems with ASP

Example

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Choice constraints:
All the subsets of {a; b; c} are candidates

Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},$$
$$y \leq 0.6$$

Hybrid problems with ASP

Example

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Choice constraints:
All the subsets of {a; b; c} are candidates

Optimisation variable domains

Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},$$
$$y \leq 0.6$$

Hybrid problems with ASP

Example

Hybrid constraints:

$$0 \ \{a; b; c\} \ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Choice constraints:
All the subsets of {a; b; c} are candidates

Objective function

Optimisation variable domains

Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},$$
$$y \leq 0.6$$

Hybrid problems with ASP

Example

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$

$$\max \quad y.$$

$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$

$$\text{with } x, y \in \mathbb{R}^+$$

Choice constraints:
All the subsets of {a; b; c} are candidates

Objective function

Hybrid constraints:

$$y \geq 1 \leftarrow a.$$

If ***a*** is true, then ***y* ≥ 1** should be true

Optimisation variable domains

Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},$$
$$y \leq 0.6$$

Hybrid problems with ASP

Example

Hybrid constraints:

$0\ \{a; b; c\}\ 3.$

$\max \quad y.$

$y \geq 1 \leftarrow a.$
$x + y \leq 1 \leftarrow b.$
$-x + y \leq 0 \leftarrow c.$

with $x, y \in \mathbb{R}^+$

Choice constraints:
All the subsets of {a; b; c} are candidates

Objective function

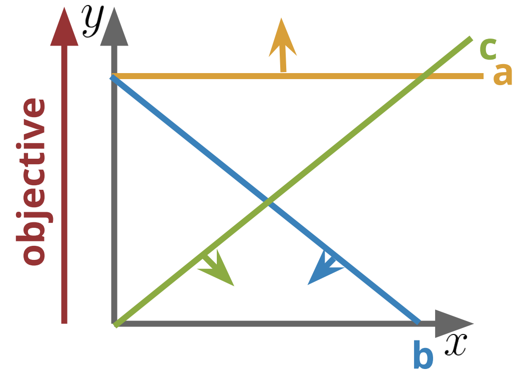
Hybrid constraints:

$y \geq 1 \leftarrow a.$

If ***a*** is true, then ***y* ≥ 1** should be true

Optimisation variable domains

Visual representation of the LP problem



Integrity constraints:

$\forall x, y \in \text{LP-Solutions},$
$y \leq 0.6$

Hybrid problems with ASP

Example

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

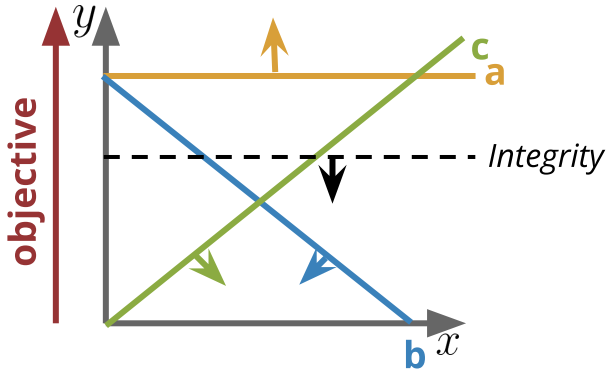
Choice constraints:
All the subsets of {a; b; c} are candidates

Objective function

Hybrid constraints:
$y \geq 1 \leftarrow a.$
If ***a*** is true, then ***y*** **≥ 1** should be true

Optimisation variable domains

Visual representation of the LP problem



Integrity constraints:

$$\forall x, y \in \text{LP-Solutions},\ y \leq 0.6$$
$$\equiv$$
$$\max \quad y \leq 0.6$$

2-QBF formulas over reals

Two levels of Boolean quantifiers [1]*:*

Given a set of optimisation constraints, there is no real valid solutions such that ***y*** **≤ 0.6**

[1] 2-QBF formulas over Boolean are Σ_2^P-complete — T. Eiter et al., **AMAI**, 1995

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* — *example: z3[1] (SAT + LP), DPLL extension*

2. *ASP modulo theory* — *example: clingoLP[2] (ASP + LP solver)*

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* — *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* — *example: clingoLP[2] (ASP + LP solver)*

[1] L. de Moura et al., **TACAS**, 2008

[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* *example: z3*[1] *(SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* *example: clingoLP*[2] *(ASP + LP solver)*
   → *Integrity constraints* are not natively support, but can be extended

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* *example: clingoLP[2] (ASP + LP solver)*
   → *Integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$0\ \{a; b; c\}\ 3.$
$\max\quad y.$
$y \geq 1 \leftarrow a.$
$x + y \leq 1 \leftarrow b.$
$-x + y \leq 0 \leftarrow c.$
$\text{with } x, y \in \mathbb{R}^+$

Integrity constraints:

$\max\quad y \leq 0.6$

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* *example: clingoLP[2] (ASP + LP solver)*
   → *integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Integrity constraints:

$$\max \quad y \leq 0.6$$

Candidate solutions:

$\{\}$
$\{a\}$
$\{b\}$
$\{c\}$
$\{a; b\}$
$\{a; c\}$
$\{b; c\}$
$\{a; b; c\}$

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* — *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* — *example: clingoLP[2] (ASP + LP solver)*
   → *integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max \quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Integrity constraints:

$$\max \quad y \leq 0.6$$

Candidate solutions:

$\{\}$
$\{a\}$
$\{b\}$
$\{c\}$
$\{a; b\}$
$\{a; c\}$
$\{b; c\}$
~~$\{a; b; c\}$~~ **No real solutions**

1. **Enumerate all the stable models**
   Do not consider integrity constraints

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* — *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* — *example: clingoLP[2] (ASP + LP solver)*
   → *integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$$0\ \{a; b; c\}\ 3.$$
$$\max\quad y.$$
$$y \geq 1 \leftarrow a.$$
$$x + y \leq 1 \leftarrow b.$$
$$-x + y \leq 0 \leftarrow c.$$
$$\text{with } x, y \in \mathbb{R}^+$$

Integrity constraints:

$$\max\quad y \leq 0.6$$

Candidate solutions:

$\{\} \rightarrow \max\ \ y = \infty$

$\{a\} \rightarrow \max\ \ y = \infty$

$\{b\} \rightarrow \max\ \ y = 1$

$\{c\} \rightarrow \max\ \ y = \infty$

$\{a; b\} \rightarrow \max\ \ y = 1$

$\{a; c\} \rightarrow \max\ \ y = \infty$

$\{b; c\} \rightarrow \max\ \ y = 0.5$

~~$\{a; b; c\}$~~ **No real solutions**

1. **Enumerate all the stable models**
   Do not consider integrity constraints
2. **Compute optimal solution**
   Compute the optimum for each stable model

[1] L. de Moura et al., **TACAS**, 2008

[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* *example: clingoLP[2] (ASP + LP solver)*
   → *integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$0\ \{a; b; c\}\ 3.$
$\max \quad y.$
$y \geq 1 \leftarrow a.$
$x + y \leq 1 \leftarrow b.$
$-x + y \leq 0 \leftarrow c.$
$\text{with } x, y \in \mathbb{R}^+$

Integrity constraints:

$\max \quad y \leq 0.6$

Candidate solutions:

~~$\{\} \rightarrow \max \ y = \infty$~~
~~$\{a\} \rightarrow \max \ y = \infty$~~
~~$\{b\} \rightarrow \max \ y = 1$~~
~~$\{c\} \rightarrow \max \ y = \infty$~~
~~$\{a; b\} \rightarrow \max \ y = 1$~~
~~$\{a; c\} \rightarrow \max \ y = \infty$~~
$\{b; c\} \rightarrow \max \ y = 0.5$
~~$\{a; b; c\}$~~ **No real solutions**

1. **Enumerate all the stable models**
   Do not consider integrity constraints
2. **Compute optimal solution**
   Compute the optimum for each stable model
3. **Filter all the solution which do not respect integrity constraints**

[1] L. de Moura et al., **TACAS**, 2008
[2] T. Janhunen et al., **TPLP**, 2017

Solving hybrid problems

State of the art

Several existing approach to solve hybrid problem

Based on:

1. *Satisfiability modulo theory* — *example: z3[1] (SAT + LP), DPLL extension*
   → Could not solve optimisation problem
2. *ASP modulo theory* — *example: clingoLP[2] (ASP + LP solver)*
   → *integrity constraints* are not natively support, but can be extended

example:

Hybrid constraints:

$0\ \{a; b; c\}\ 2$

$\max \quad y.$

$y \geq 1 \leftarrow$

$x + y \leq$

$-x + y \leq$

with x, y

Integrity constraints:

$\max \quad y \leq 0.6$

Candidate solutions:

~~$\{\} \rightarrow \max\ y = \infty$~~

~~$\{a; b; c\}$~~ **No real solutions**

1. **Enumerate all the stable models**

...ints

...stable

...do not respect integrity constraints

Enumerating all the solution is too **costly**

Too many valid stable models, too many calls to the optimisation solvers, etc.

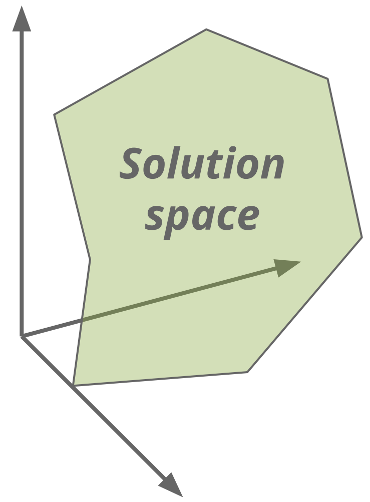
More efficient approaches are needed !

[1] L. de Moura et al., **TACAS**, 2008

[2] T. Janhunen et al., **TPLP**, 2017

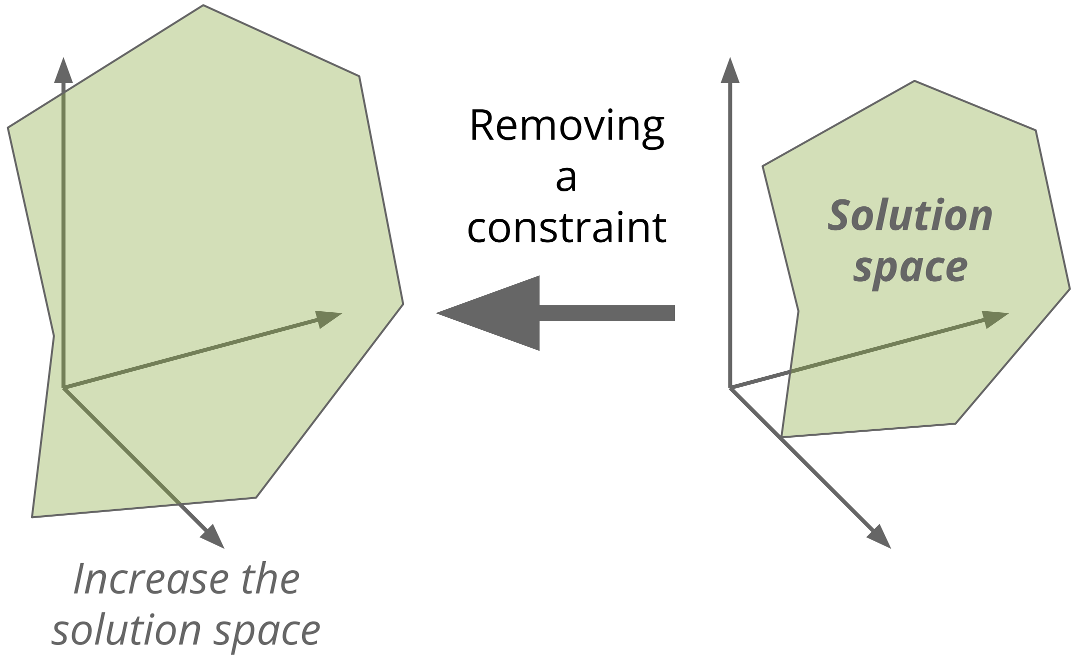
Optimisation problem properties

For satisfiability



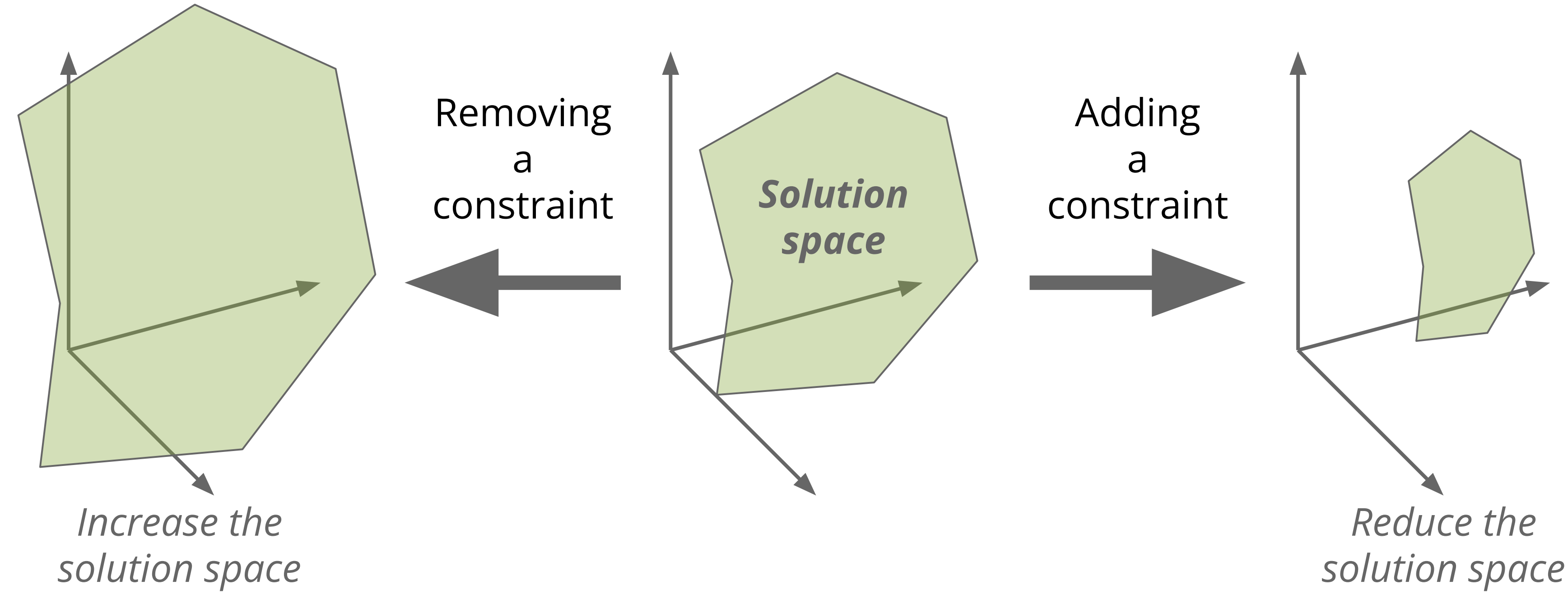
Optimisation problem properties

For satisfiability



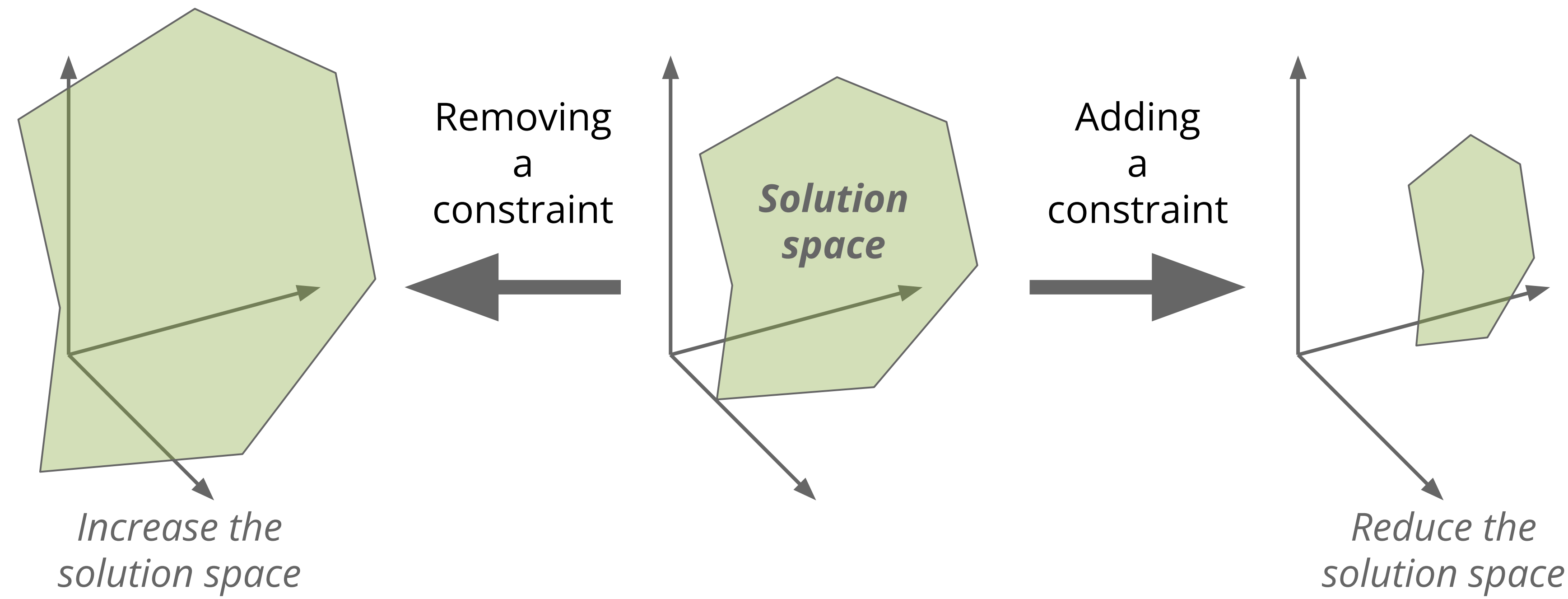
Optimisation problem properties

For satisfiability



Optimisation problem properties

For satisfiability



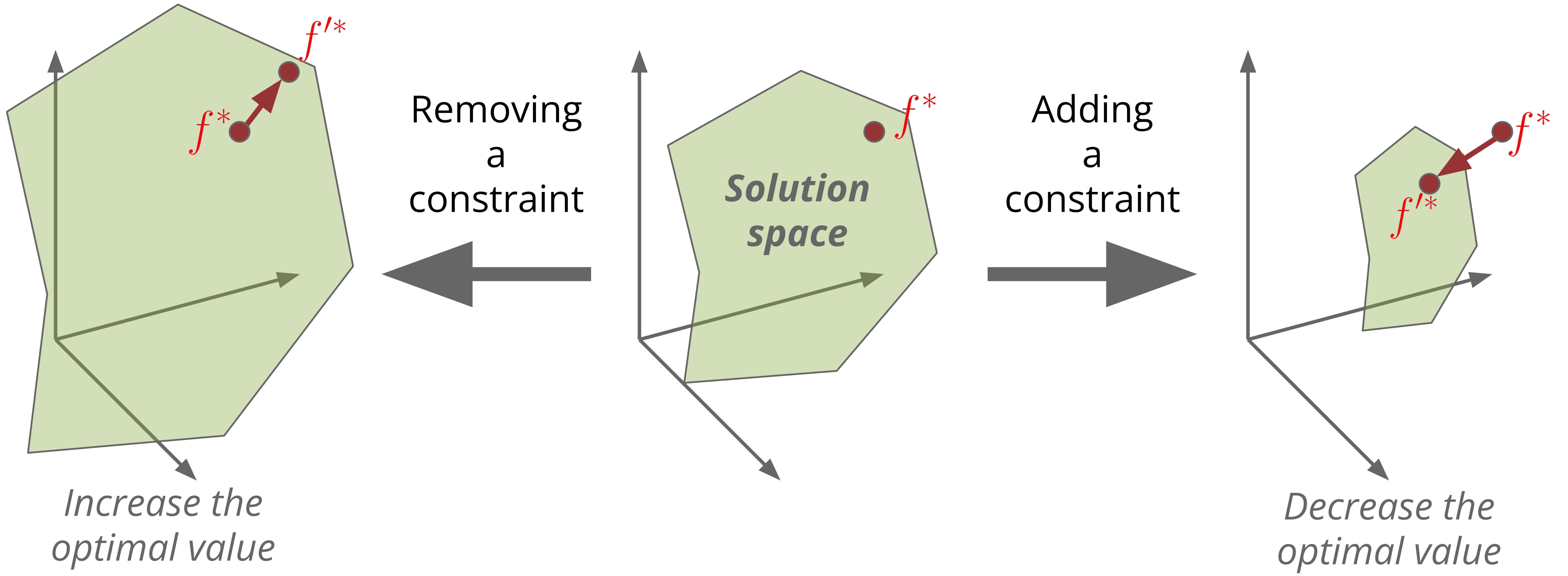
Monotone property (satisfiability)

Adding constraints to an UNSAT problem → **UNSAT**
Removing constraints from a SAT problem → **SAT**

Already considered to compute Irreducible Infeasible Set in hybrid solvers

Optimisation problem properties

For optimum



Monotone property (optimal value)

Given an optimal value f^* to an optimisation problem,
Adding a constraints $\rightarrow f'^* \geq f^*$
Removing a constraints $\rightarrow f'^* \leq f^*$

Monotone property

Example

Optimisation constraints subsets can be partially ordered

Hybrid problem: ASP + LP

~~$0\ \{a; b; c\}\ 3.$~~

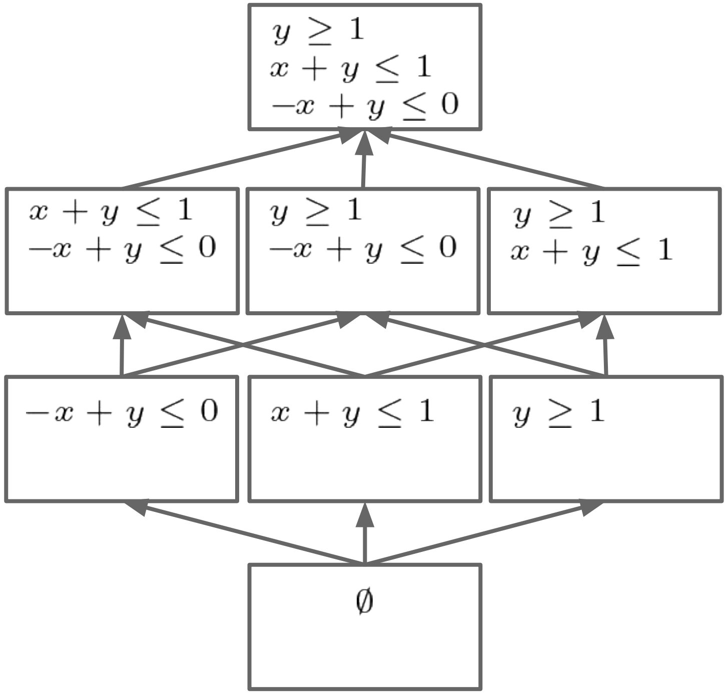
$\max \quad y.$

$y \geq 1 \leftarrow a.$

$x + y \leq 1 \leftarrow b.$

$-x + y \leq 0 \leftarrow c.$

with $x, y \in \mathbb{R}^+$



Hasse diagram: all the constraints subsets

Monotone property

Example

Optimisation constraints subsets can be partially ordered

Hybrid problem: ASP + LP

~~$0\ \{a; b; c\}\ 3.$~~

$\max \quad y.$

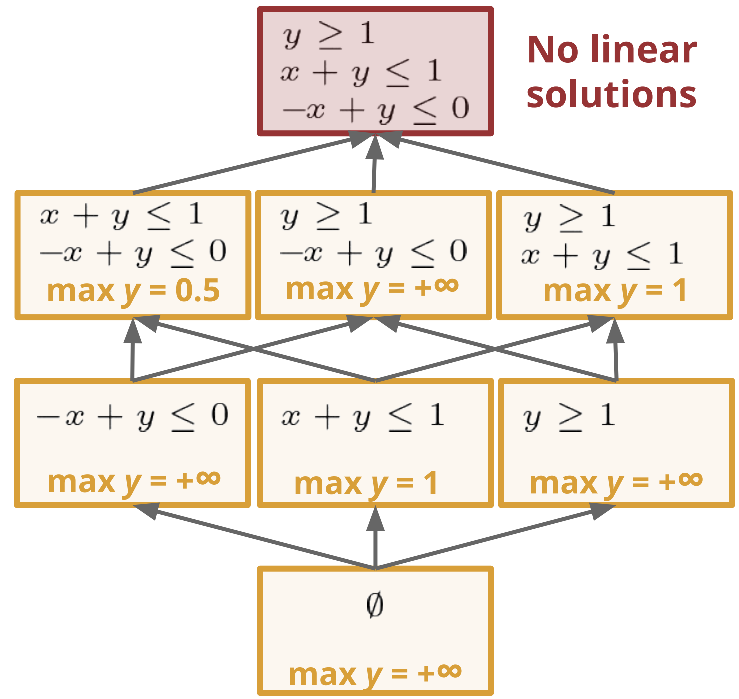
$\boxed{y \geq 1} \leftarrow a.$

$\boxed{x + y \leq 1} \leftarrow b.$

$\boxed{-x + y \leq 0} \leftarrow c.$

with $x, y \in \mathbb{R}^+$

Compute the optimal solution → **for each constraint subsets**



Hasse diagram: all the constraints subsets

Can be extended to define equivalence classes of optimal problems

Monotone property

Example

Optimisation constraints subsets can be partially ordered

Hybrid problem: ASP + LP

~~$0\ \{a; b; c\}\ 3.$~~

$\max \quad y.$

$\boxed{y \geq 1} \leftarrow a.$

$\boxed{x + y \leq 1} \leftarrow b.$

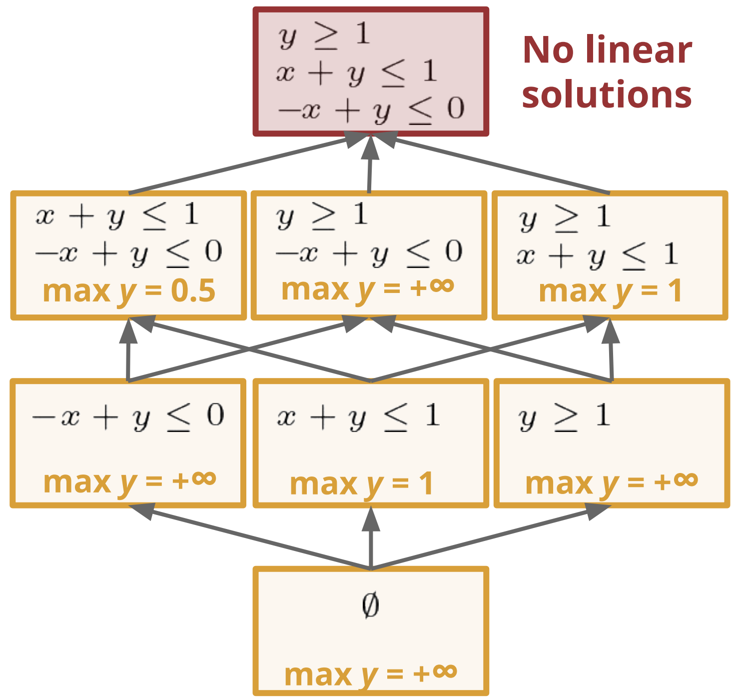
$\boxed{-x + y \leq 0} \leftarrow c.$

with $x, y \in \mathbb{R}^+$

Compute the optimal solution

→

for each constraint subsets



Hasse diagram: all the constraints subsets

We can **deduce knowledge** from one sets of constraints to **all its subsets and supersets**

Can be extended to define equivalence classes of optimal problems

Merging ASP and optimisation constraints

From optimisation constraints to literals

Associating a literal l_c to each constraint c such that:
l_c is true (1) iff the constraint c is considered

example:

Hybrid problem: ASP + LP

$$
\begin{aligned}
&0\ \{a; b; c\}\ 3.\\
&\max \quad y.\\
&y \geq 1 \leftarrow a.\\
&x + y \leq 1 \leftarrow b.\\
&-x + y \leq 0 \leftarrow c.\\
&\text{with } x, y \in \mathbb{R}^+
\end{aligned}
$$

Merging ASP and optimisation constraints

From optimisation constraints to literals

Associating a literal l_c to each constraint c such that:

l_c is true (1) iff the constraint c is considered

example:

Hybrid problem: ASP + LP

$0\ \{a; b; c\}\ 3.$

$\max \quad y.$

$y \geq 1 \leftarrow a.$

$x + y \leq 1 \leftarrow b.$

$-x + y \leq 0 \leftarrow c.$

$\text{with } x, y \in \mathbb{R}^+$

Replace hybrid theory atoms

$l_a := y \geq 1$

$l_b := x + y \leq 1$

$l_c := -x + y \leq 0$

Problem: ASP

$0\ \{a; b; c\}\ 3.$

$\max \quad y.$

$l_a \leftarrow a.$

$l_b \leftarrow b.$

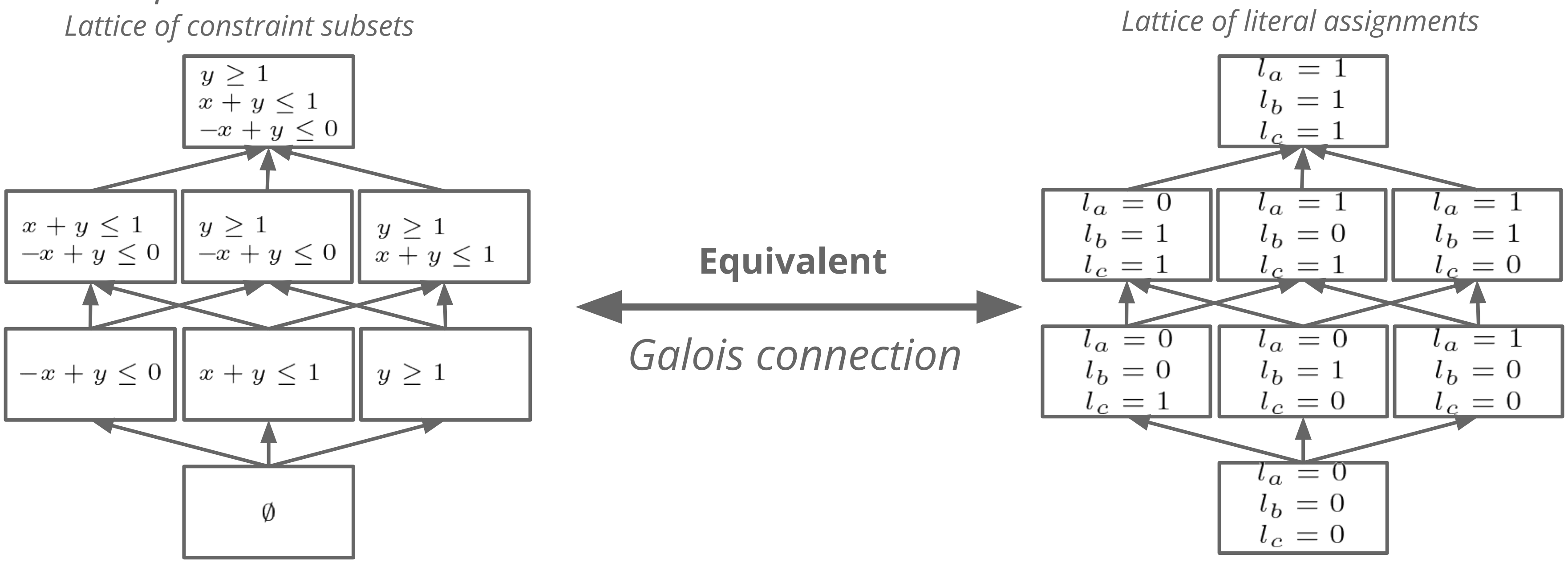
$l_c \leftarrow c.$

Merging ASP and optimisation constraints

From optimisation constraints to literals

Associating a literal l_c to each constraint c such that:
l_c is true (1) iff the constraint c is considered

example:



All the monotone properties are conserved

Improving search space exploration

Constraint propagation

Generalisation of counter-example to not check solutions that will not satisfy the integrity constraints

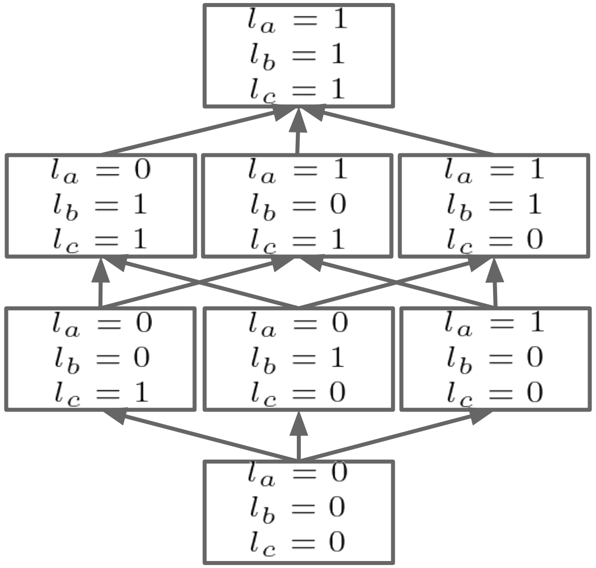
example:

Integrity constraint:

$$\max \quad y \leq 0.6$$

Resolution:

Lattice of literal assignments



Improving search space exploration

Constraint propagation

Generalisation of counter-example to not check solutions that will not satisfy the integrity constraints

example:

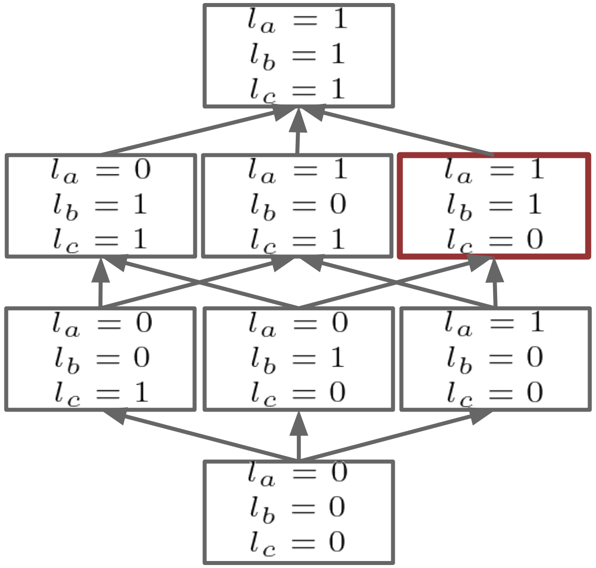
Integrity constraint:

$$\max \quad y \leq 0.6$$

Resolution:

1. $\begin{matrix} l_a = 1 \\ l_b = 1 \\ l_c = 0 \end{matrix} \longrightarrow \max \quad y = 1$

Lattice of literal assignments



Improving search space exploration

Constraint propagation

Generalisation of counter-example to not check solutions that will not satisfy the integrity constraints

example:

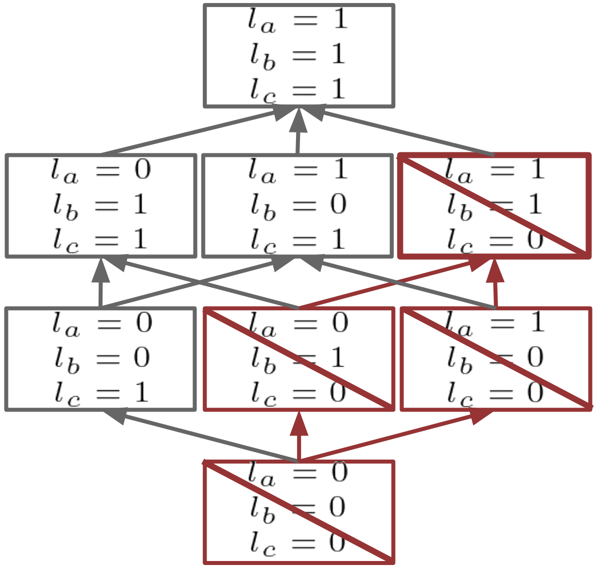
***Integrity constraint**:*

$$\max \quad y \leq 0.6$$

***Resolution**:*

1. $\begin{matrix} l_a = 1 \\ l_b = 1 \\ l_c = 0 \end{matrix} \rightarrow \max \quad y = 1$

Lattice of literal assignments



Prohibited all subsets

All subset will have an optimum ≥ 1

Improving search space exploration

Constraint propagation

Generalisation of counter-example to not check solutions that will not satisfy the integrity constraints

example:

Integrity constraint:

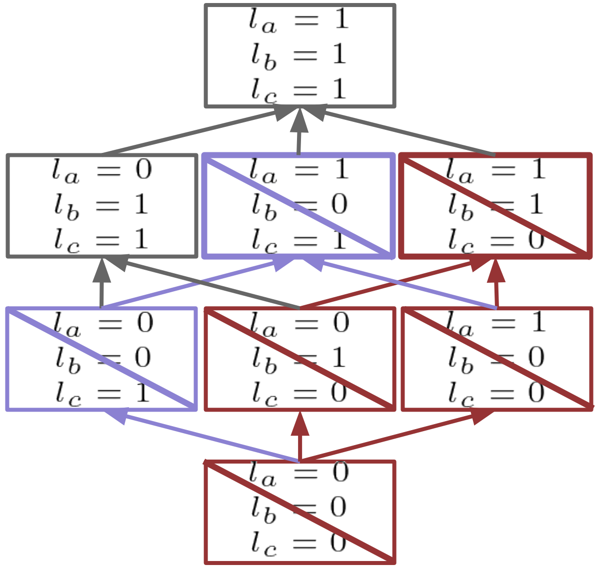
$$\max \quad y \leq 0.6$$

Resolution:

1. $\begin{matrix} l_a = 1 \\ l_b = 1 \\ l_c = 0 \end{matrix} \longrightarrow \max \quad y = 1$

2. $\begin{matrix} l_a = 1 \\ l_b = 0 \\ l_c = 1 \end{matrix} \longrightarrow \max \quad y = \infty$

Lattice of literal assignments



Prohibited all subsets

All subset will be ∞

Improving search space exploration

Constraint propagation

> Generalisation of counter-example to not check solutions that will not satisfy the integrity constraints

example:

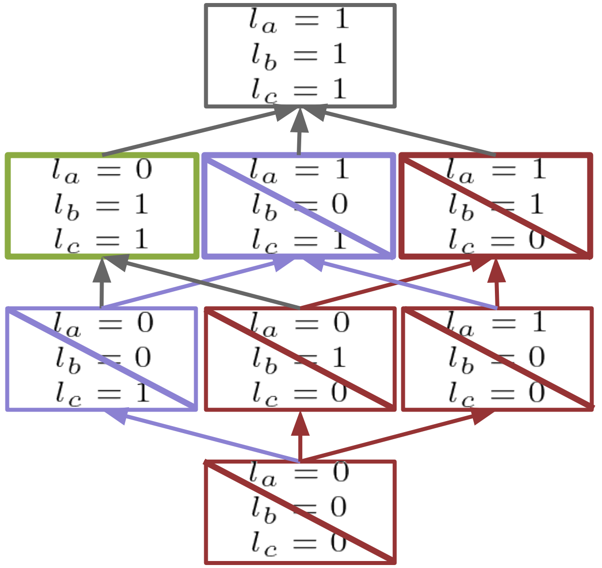
Integrity constraint:

$$\max \quad y \leq 0.6$$

Resolution:

1. $\begin{matrix} l_a = 1 \\ l_b = 1 \\ l_c = 0 \end{matrix} \rightarrow \max \quad y = 1$
2. $\begin{matrix} l_a = 1 \\ l_b = 0 \\ l_c = 1 \end{matrix} \rightarrow \max \quad y = \infty$
3. $\begin{matrix} l_a = 0 \\ l_b = 1 \\ l_c = 1 \end{matrix} \rightarrow \max \quad y = 0.5$

Lattice of literal assignments



A solution is found

Implementation with clingo in practice

Rely on python API of clingo[1] and its propagator interface[2]

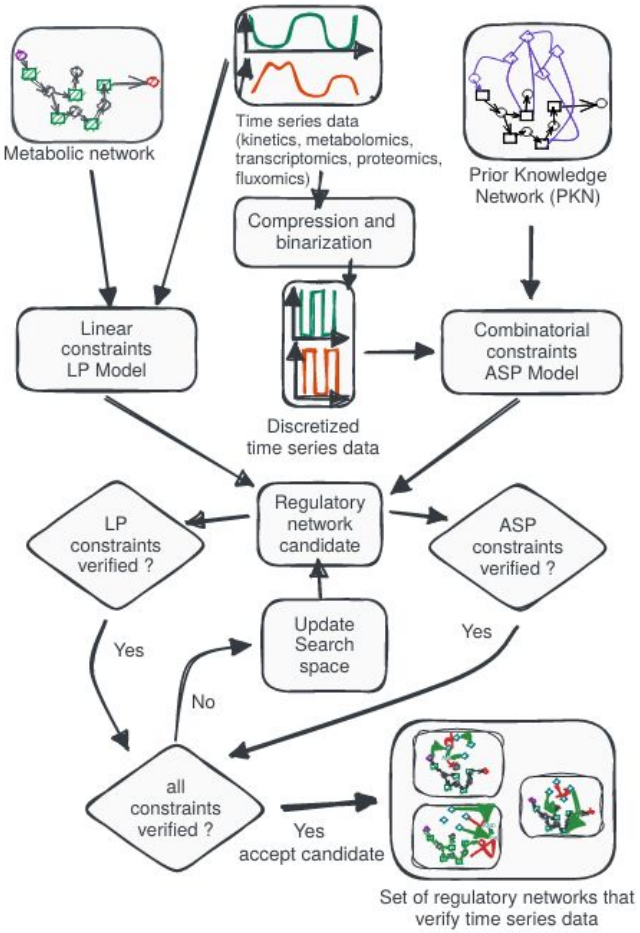
Rely on 4 functions:

Initialize	**Undo**	**Propagate**	**Check**
1. Associate a literal to each optimisation constraints 2. Initialise the data-structures in memory	*Backtrack the literals affectation* Remove backtracked literals values from memory	*Optimisation literals have been assigned* Update the memory with assigned literals values	*All the optimisation literals have been assigned* 1. Solve the optimisation problem with activated constraints 2. Accept/Reject solutions according satisfying *integrity constraints* 3. Add new constraints
Beginning of the solving process	*Conflict resolution*	*Literals are assigned* ***and*** *All literals are not assigned*	*All literals are assigned*

Call

[1] M. Gebser et al., **TPLP**, 2019
[2] R. Kaminski et al., **ArXiv**, 2021

Application example: MERRIN



Bioinformatics problem:

Learning regulatory rules from metabolic traces

Hybrid problem:

- Combinatorial:
  Search space of admissible regulatory rules defined by combinatorial rules
- Linear:
  *Simulation of cell's metabolism with **FBA***

Conclusion

Solving hybrid problem with integrity constraints over reals

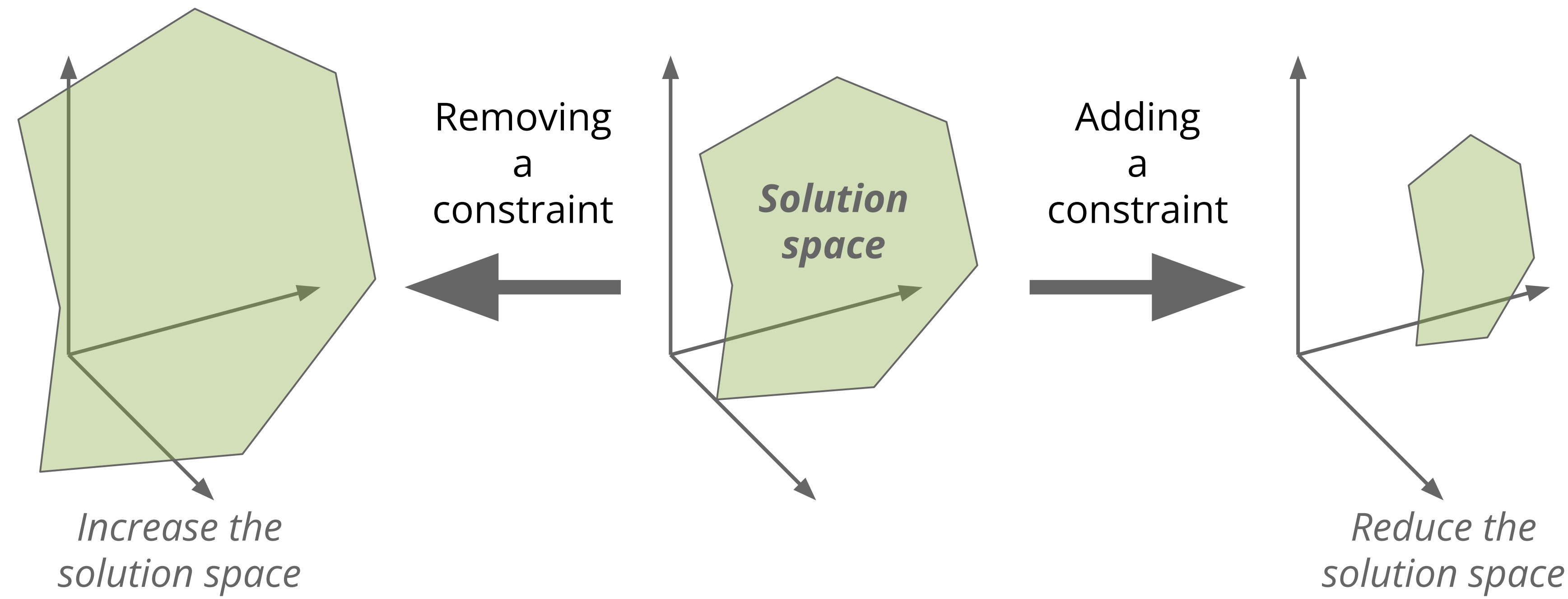
- Monotone properties on optimisation problem states
  Over the problem state (e.g bound, sat, etc.) and optimum values
- Implementable with *clingo* (ASP solver)
  Currently a problem specific implementation: MERRIN
  Can be used to do optimisation over reals

Future works

- Generic implementation and benchmarks
- Lattice element traversal heuristics
  Guiding ASP resolution to efficiently traverse the lattice
  When should we check the state of the optimisation problem?
- Efficient data structure to model the lattice

Optimisation problem properties

For satisfiability



Monotone property (satisfiability)

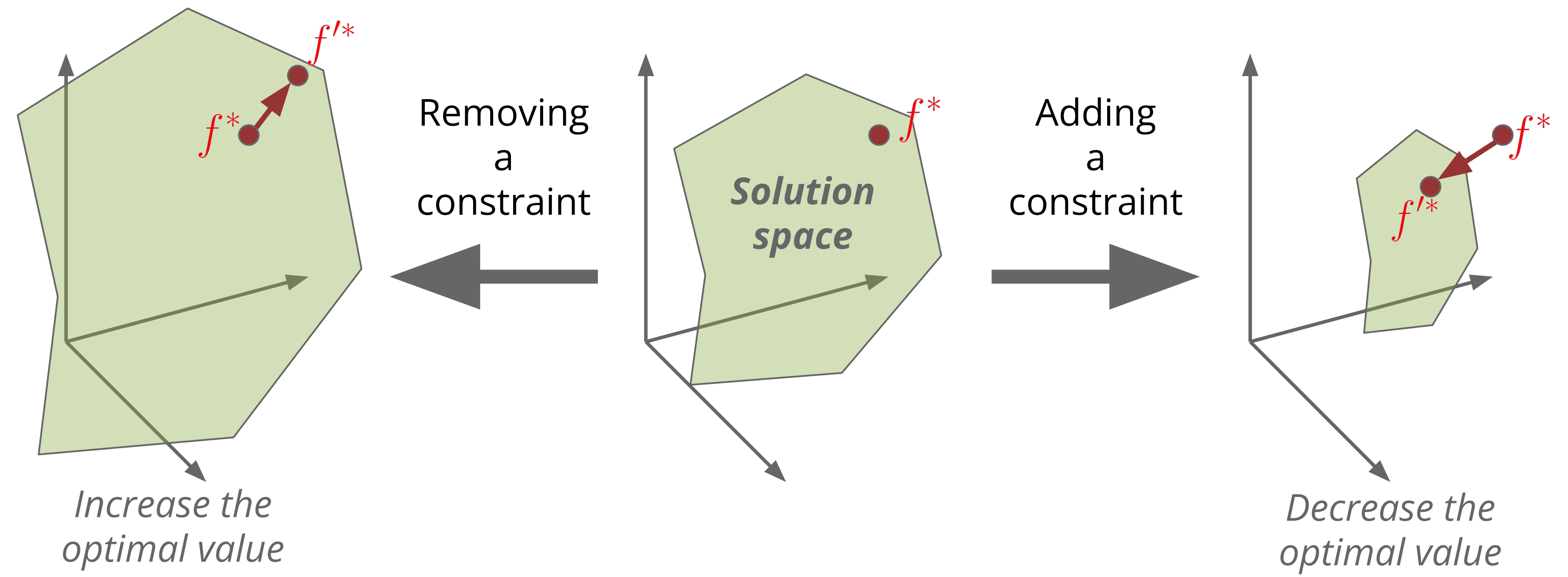
C_1, C_2 : sets of constraints

$$\text{UNSAT}(C_1) \wedge C_1 \subseteq C_2 \implies \text{UNSAT}(C_2)$$

Already considered to compute Irreducible Infeasible Set in hybrid solvers

Optimisation problem properties

For optimum



Monotone property (optimal value)

C_1, C_2 : sets of constraints f_1, f_2 : their optimal values

$$C_1 \subseteq C_2 \implies f_1 \geq f_2$$

Monotone property on lattice

Example

Partial ordered set of the set of all constraint subsets

Hybrid problem: ASP + LP

$0\ \{a; b; c\}\ 3.$
$\max\quad y.$
$y \geq 1 \leftarrow a.$
$x + y \leq 1 \leftarrow b.$
$-x + y \leq 0 \leftarrow c.$
$\text{with } x, y \in \mathbb{R}^+$

1. **Extracting the optimisation problem:**
   Objective function: $\max\quad y$
   Variable domains: $\text{with } x, y \in \mathbb{R}^+$
   Constraints:

$$y \geq 1$$
$$x + y \leq 1$$
$$-x + y \leq 0$$

Monotone property on lattice

Example

$\max \quad y$

$\text{with } x, y \in \mathbb{R}^+$

$$\begin{array}{l} y \geq 1 \\ x + y \leq 1 \\ -x + y \leq 0 \end{array}$$

$$\begin{array}{l} x + y \leq 1 \\ -x + y \leq 0 \end{array} \qquad \begin{array}{l} y \geq 1 \\ -x + y \leq 0 \end{array} \qquad \begin{array}{l} y \geq 1 \\ x + y \leq 1 \end{array}$$

$$-x + y \leq 0 \qquad x + y \leq 1 \qquad y \geq 1$$

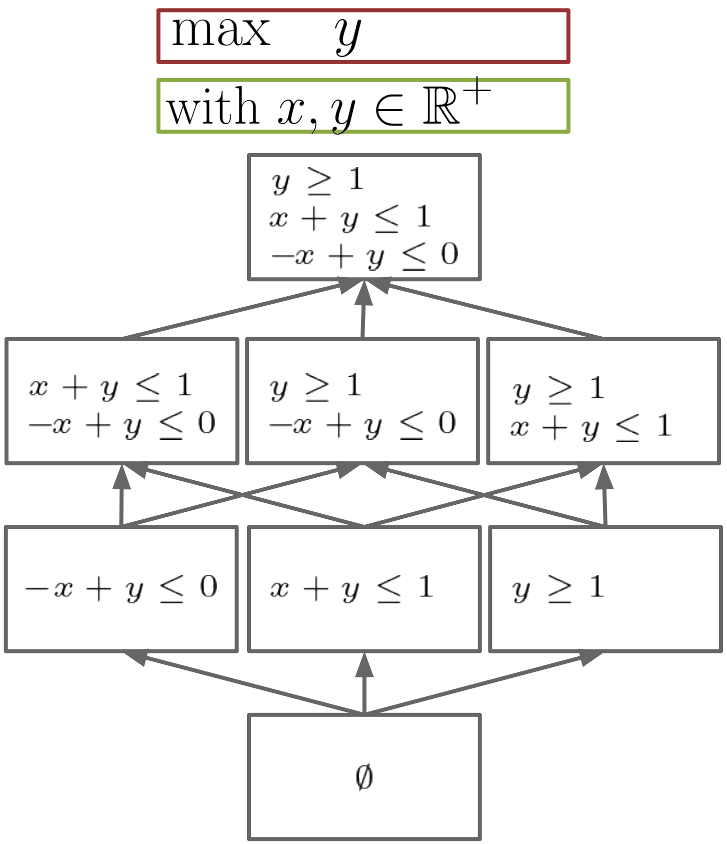
$$\emptyset$$

Partial ordered set of the set of all constraint subsets

1. **Extracting the optimisation problem:**
   Objective function: $\max \quad y$
   Variable domains: $\text{with } x, y \in \mathbb{R}^+$
   Constraints:
   $$\begin{array}{l} y \geq 1 \\ x + y \leq 1 \\ -x + y \leq 0 \end{array}$$
2. **Compute all the subsets of constraints**
   8 subsets of constraints

Monotone property on lattice

Example



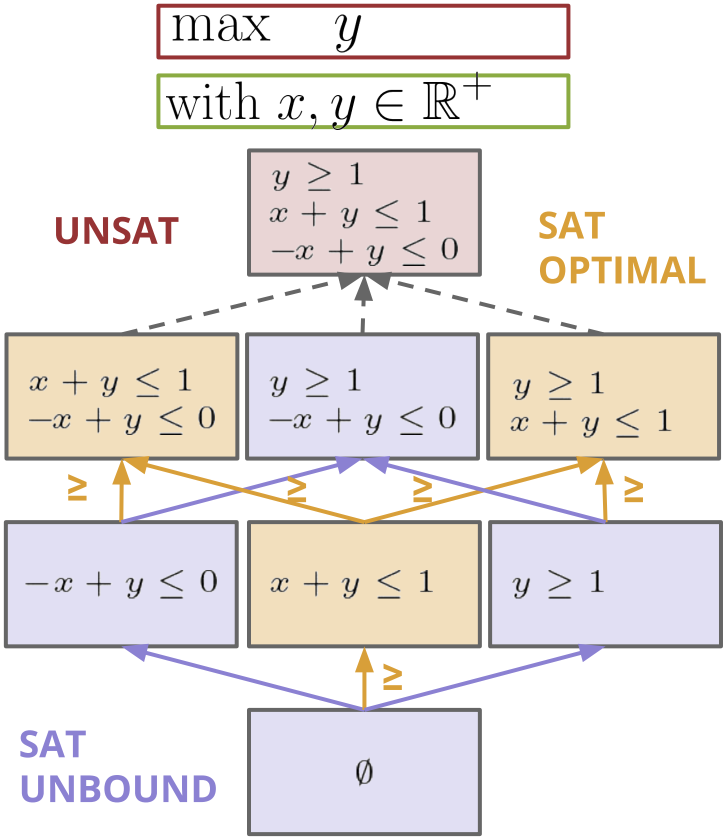
Hasse diagram

Partial ordered set of the set of all constraint subsets

1. **Extracting the optimisation problem:**
   Objective function: $\max \quad y$
   Variable domains: $\text{with } x, y \in \mathbb{R}^+$
   Constraints:
   $$\begin{array}{l} y \geq 1 \\ x + y \leq 1 \\ -x + y \leq 0 \end{array}$$
2. **Compute all the subsets of constraints**
   8 subsets of constraints
3. **Ordered the constraints subsets with inclusion**

Monotone property on lattice

Definition



Hasse diagram

Partial ordered set of the set of all constraint subsets

C_1, C_2 : sets of constraints
f_1, f_2 : their optimal values

Monotone Property

$$\text{UNSAT}(C_1) \wedge C_1 \subseteq C_2 \implies \text{UNSAT}(C_2)$$
$$C_1 \subseteq C_2 \implies f_1 \geq f_2$$

We can thus **deduce knowledge** from one sets of constraints to **all its subsets and supersets**

Can be extended to define equivalence classes of optimal problems